

Applicant(s): JAMES A. STARKEY
Serial No.: 09/919,052
Filed: July 31, 2001

In the Specification

Please replace the paragraph beginning at page 2, line 22 as follows:

U. S. Patent No. 5,987,480 to Donohoue et al. depicts an example of a second approach sometimes referred to as a "active page server" tool. In accordance with this approach, a web page server stores the HTML, or other code required for generating a web page, such as `[[HDML]]HTML` code, with all the header and body information as a complete web page or template. These pages include the HTML code that controls the appearance of the web page and other code that controls web page semantics. In the Donohoue et al. patent this other code uses "@" as a symbol that defines paired delimiters. During the processing of a particular web page or template, a web server locates these symbols or delimiters. The web server then interprets each character string between paired delimiters. In the Donohoue et al. patent, these character strings can define a dynamic tag, an "if" instruction or a "loop" instruction. If the character string defines a dynamic tag, the value generated by the dynamic tag replaces the corresponding paired delimiters and dynamic tag. If the character string defines an "if" instruction, the system performs a defined action if a defined condition is met. If a "loop" instruction is included, all the values during each loop operation replace the paired delimiters and "loop" instruction. Once all this information is complete,

Applicant(s): JAMES A. STARKEY
Serial No.: 09/919,052
Filed: July 31, 2001

the web server sends the HTML code with the embedded values to the browser. As with the first approach, a web page or template in an active page system also establishes web page semantics.

Please replace the paragraph beginning at page 9, line 20 as follows:

The web page generator 20, may include multiple application servers, such as the application server 24. It also includes a web server 30, a library 31, a local directory 32 and a front end module ~~[[34]]~~33. The web server 30 can access information in the library 31 and, in some cases, the local directory 32, to select a specific application server, such as the application server 24, in response to a browser request. When an application server generates the HTML or other code, the front end module 33 and web server 30 transfer the code to the user's browser. Information stored in the library 31 and local directory 32 may also be used during such transfers.

Applicant(s): JAMES A. STARKEY
Serial No.: 09/919,052
Filed: July 31, 2001

Please replace the paragraph beginning at page 10, line 6 as follows:

More specifically, when a user, such as USERm 22, desires to communicate with a web page generator, the USERm 22 through a browser generates a user request having any standard form, such as "www.domainname", and including other information such as the user's address and browser type, as known. Referring now to FIGS. 1 and 2, the web server 30 receives that user request in step 34. Step 35 translates that user request into a modified form, specifically:

www.domainname/WEB.NFS?a=applicationname&s= session_id
wherein everything to the right of the "?" constitutes a query string. Step 35 transfers this modified request to the front end module 33. In step 36 the front end module 33 uses the modified request to attach to an application server containing the named application, the application server selection being based on the information [to the left]to the left of the "?".

Please replace the paragraph beginning at page 18, line 6 as follows:

FIG. 4 depicts representative templates from the templates store 25 in the content store 26 including, in a base layer, a DrlPage template 100 and DrlTable template 101. FIG. 5A depicts the content of the DrlPage template 100. It includes a page header template at 102 and invokes the DrlTable template

Applicant(s): JAMES A. STARKEY
Serial No.: 09/919,052
Filed: July 31, 2001

101 at 103. The DrlTable template 101 in FIG. 5B specifies a border at 104 and provides a format at 105 for displaying columns as individual records in a result set are processed on an iterative basis. The statements at 106 are special statements that control the update of a current record in a result set. The body of the DrlPage template ends with a page trailer template 107. As will now be apparent, this structure allows one template to "call" another template. In fact one template can `[[all]]call` any of several templates directly or indirectly (e.g., Template 1 calls Template 2 that, in turn, calls Template 3 and Template 4.)

Please replace the paragraph beginning at page 25, line 15 as follows:

If the template name has not been used previously, step 167 transfers control to step 174 that places the new Template Object at the head of the collision chain. If the template being analyzed does not have the same name as another template in the collision chain, step 171 bypasses steps 172 through 173 to enable the operation in step 174. When control reaches step 175 from steps 173 or 174, step 175 determines if any additional templates remain in the result set. If additional templates exist, control passes back to step `[[163]]162`. Otherwise this portion of the Template Set Object method is complete and control transfers to FIG. 8D.

Applicant(s): JAMES A. STARKEY
Serial No.: 09/919,052
Filed: July 31, 2001

Please replace the paragraph beginning at page 28, line 10 as follows:

When the processes of FIGS. 8B or 8C-1 and 8C-2 are completed, a hierarchy of all the templates for the application exists. Application link slots, such as the application link slot 138 in the Template Set Object of FIG. 6C, and the application [[link]]chain slot in each Template Object in the same layer, such as the application [[link]]chain slot 146 in FIG. 6D, link all of the Template Objects for a given layer. Each collision chain links each Template Object with templates names that generate the same hash code. If one or more layers have multiple Template Objects with the same name, the collision chain identifies a first one of the Template Objects. The homonym chain identifies any other Template Objects in the same layer and all other Template Objects in lower levels. Consequently, the homonym chain for a given Template Set Object identifies all the Template Objects with the same name at that application layer and any lower layer (i.e., that base layer, the application layer and any intervening application layers).

Please replace the paragraph beginning at page 40, line 5 as follows:

Another advantage appears when a developer completes a web site development and packages that site for development. The template hierarchy and included relational data base simplify